DOCUMENT RESUME

ED 336 067 IR 015 074

AUTHOR Corbett, Albert T.; Anderson, John R.
TITLE Feedback Control and Learning To Program with the CMU Lisp Tutor.
PUB DATE Apr 91
NOTE 11p.; Paper presented at the Annual Meeting of the American Educational Research Association (72nd, Chicago, IL, April 3-7, 1991)
PUB TYPE Reports - Research/Technical (143) -- Speeches/Conference Papers (150) -- Tests/Evaluation Instruments (160)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Comparative Analysis; *Computer Assisted Instruction; *Feedback; Higher Education; Instructional Effectiveness; *Intermode Differences; Pretests Posttests; Problem Solving; *Programing; Questionnaires; Student Attitudes; Time on Task; *Tutorial Programs
IDENTIFIERS Paper and Pencil Tests

ABSTRACT

This study manipulated the timing and control of error feedback in problem solving and examined their effects on skill acquisition by 40 undergraduate students learning to program in the computer language Lisp under four error feedback conditions. These four conditions included two types of symbol-by-symbol feedback that vary in content, a condition in which feedback is presented only upon request as the student works, and a condition in which no feedback is available until the student has completed an exercise. Questions of interest in the study were whether: (1) the feedback manipulation affects posttest performance and the timecourse of learning; (2) the feedback manipulation affects students' perceptions of how well they have learned the material; and (3) students have preferences among feedback conditions. Results indicated that students who received feedback learned better than students in the no feedback condition, and that students in the two immediate feedback conditions finished the exercises most quickly. (8 references) (DB)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☐ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

ED336067

Feedback Control and Learning To Program with the CMU Lisp Tutor

Albert T. Corbett

John R. Anderson

Paper presented at the Annual Meeting of the American Educational Research Association (72nd, Chicago, IL, April 3-7, 1991)

IR015074

Psychology Department
Carnegie Mellon University

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Albert T. Corbett

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

BEST COPY AVAILABLE

Feedback Control and Learning to Program with the CMU Lisp Tutor

Albert T. Corbett
John R. Anderson

Psychology Department
Carnegie Mellon University

This study manipulates the timing and control of error feedback in problem solving and examines the effects on skill acquisition. More specifically, the study examines the effects of four error feedback conditions on students learning to program in the computer language Lisp. These four conditions include two-types of symbol-by-symbol feedback that vary in content, a condition in which feedback is presented only upon request as the student works and finally, a condition in which no feedback is available until the student has completed an exercise. We are interested in several issues: (1) whether the feedback manipulation affects posttest performance and the timecourse of learning (2) whether it affects students perceptions of how well they have learned the material and (3) whether students have preferences among feedback conditions.

The CMU Lisp Tutor

Students in this study worked with the CMU Lisp Tutor. The CMU Lisp Tutor is an instructional program that assists students as they complete Lisp programming exercises. In each exercise, the tutor presents a description of a short program to write and as the student types a solution, the tutor monitors the student's performance symbol-by-symbol, providing feedback on errors and providing a correct action if the student appears to be floundering. The tutor covers the first twelve chapters of an introductory programming text and has been used to teach a Lisp programming course since 1984.

The tutor is able to provide feedback in problem solving because it contains knowledge that allows it to generate solutions step-by-step along with the student. The tutor contains a set of approximately 500 production rules for generating Lisp code collectively called the *ideal student model*. Each of these productions is an if-then rule that specifies a programming goal and actions to take to satisfy the goal. An english version of a simple arithmetic rule is:

> If the goal is to add a set of numbers
> Then code the operator +,
> and set a goal to code the addends.

The term *ideal* indicates that these productions reflect the rules we hope students will acquire in learning Lisp. The tutor also contains a *bug catalog* of incorrect production rules that represent missteps a student might take in generating code. The tutor tracks the student and responds appropriately by

comparing the student's response to correct or buggy production rules that might fire at each step, a process called *model tracing*.

The tutor was developed not only as an instructional environment but also to test the ACT* model of skill acquisition (Anderson, 1983) in a "real-life" context. In this study we are specifically interested in the content and control of feedback. An early version of ACT* suggested that immediate feedback should facilitate the proceduralization of knowledge in skill acquisition. As a result, the standard tutor presents symbol-by-symbol feedback. When an error is made, the tutor immediately interrupts, presents a feedback message, deletes the error and requires the student to try again. The tutor has proven to be an effective environment for learning to program. Students working with the tutor finish the exercises more quickly than students working on their own and perform at least as well or better on posttests (Anderson and Reiser, 1985).

There are variety of reasons, though, to question whether the tutor's immediate feedback is optimal. Immediate feedback does not uniformly enhance learning (Kulik & Kulik, 1988; Schmidt, Young, Swinnen & Shapiro, 1989; Schooler & Anderson, 1990). Second, symbol-by-symbol feedback can interrupt the firing of larger production rules that may be compiled in learning. Third, students may benefit from detecting their own errors in the course of learning. We have conducted several studies in which we've modified feedback for the first two lessons of the CMU Lisp Tutor and found little evidence that immediate feedback is superior to other feedback conditions (Corbett & Anderson, 1989, 1990; Corbett, Anderson & Patterson, 1990). In the present study we are comparing four feedback conditions over a more substantial portion of the curriculum that includes lessons on recursion, the topic that students find most challenging in learning Lisp. The four conditions are (1) standard immediate feedback and correction, as described above, (2) error flagging, (3) feedback on demand and (4) no feedback.

In the Error Flagging condition, the tutor responds immediately to feedback, but does not interrupt students. It "flags" an error by displaying it in bold on the computer screen, but does not provide any explanatory text. Students are free to go back and fix the error, go back and ask for a comment on the error, or to continue coding. The tutor can recognize alternative solutions to exercises, but only more or less optimal solutions. Thus, when an error is flagged, it is possible that the student is working on non-optimal correct code that the tutor does not recognize. In this condition (and the remaining two) the tutor will accept code that works in the end, even if it does not recognize the solution. As a result, it doesn't necessarily make sense for a student to immediately correct any error that is flagged. There are two possible advantages of this condition. First, it does not interrupt the firing of large-scale productions and second, students may benefit from generating their own explanation of errors detected by the tutor.

In the Feedback on Demand condition, the tutor takes no initiative in providing help to the student. Instead, at any time, the student can ask the tutor to check over the code. If an error is found, the tutor provides the same feedback message that the standard tutor would have presented automatically. If no error exists, the student is informed accordingly. This condition may be superior to the standard condition if students benefit from detecting their own errors.

In the No Feedback condition, no assistance is available to the students as they type their code. Instead, when a student indicates that an exercise is done, the tutor reports whether or not the answer is correct. If not, the student is allowed to keep trying. Students in all four conditions have access to a Lisp interpreter which allows them to test their code. This is the only mechanism available to students in the No Feedback condition for detecting and correcting errors. Unlike the other three conditions, students in the No Feedback condition are ultimately allowed to finish an exercise with incorrect code. If the student does give up, the tutor displays a correct solution. Thus, this condition is analogous to doing exercises with answers at the back of the book.

The Experiment

Subjects.

Forty undergraduates were recruited and paid to participate in the experiment. Each student worked with one of the four versions of the tutor. Prior programming experience and Math SAT scores were balanced across the four groups.

Curriculum.

The students worked through five tutor lessons, completing 41 exercises. These five lessons covered function calls, function definitions, conditional functions, basic recursion and advanced recursion. A sample exercise from each lesson is displayed in Table 1.

Programming Tests.

Students completed a paper and pencil programming test after the second lesson. Following the fifth lesson, the students completed programming tests in three environments: (1) paper and pencil (2) online editor and lisp interpreter with no error feedback and (3) immediate error feedback with no lisp environment. The second environment is essentially identical to the No Tutor practice environment, except that students employed a familiar screen editor rather than a structured editor. Any advantage the No Tutor group displays in this condition should be attributable to debugging skills they've acquired, rather than greater familiarity with the editor. The third environment is essentially identical to the Immediate Feedback practice environment, except that no students have access to a Lisp environment. Debugging skills can not be employed in this environment.

Questionnaires.

Students completed a questionnaire after the second lesson and again after the fifth lesson. These included two questions that assess the students' self-knowledge of the learning process and a handful of questions on their opinion of the tutor.

Table 1

Example Exercises

Lesson 1

```
(car (cdr '(horse dog cat)))
```

Lesson 2

```
(defun ends (lis)
       (cons (car lis) (last lis)))
```

Lesson 3

```
(defun classify (item)
       (cond ((null item) nil)
             ((numberp item) 'number)
             ((atom item) 'atom)
             (t 'list)))
```

Lesson 4

```
(defun fact (num)
       (cond ((zerop num) 1)
             (t (* num (fact (1- num))))))
```

Lesson 5

```
(defun delete-item (item lis)
       (cond ((null lis) nil)
             ((equal item (car lis))
               (delete-item item (cdr lis)))
             ((atom (car lis))
               (cons (car lis) (delete-item item (cdr lis))))
             (t (cons (delete-item item (car lis))
                      (delete-item item (cdr lis))))))
```

Results

Time on Task

Average time to complete an exercise in the five tutor lessons is displayed in Figure 1. Not surprisingly, students in the Immediate Feedback condition finish the exercises most quickly while students in the No Feedback condition take the longest to complete the exercises. Interestingly, students in the Error Flagging condition take slightly longer than those in the Feedback on Demand condition in the early lessons. However, students begin to benefit from immediate error flagging in the later more difficult lessons.

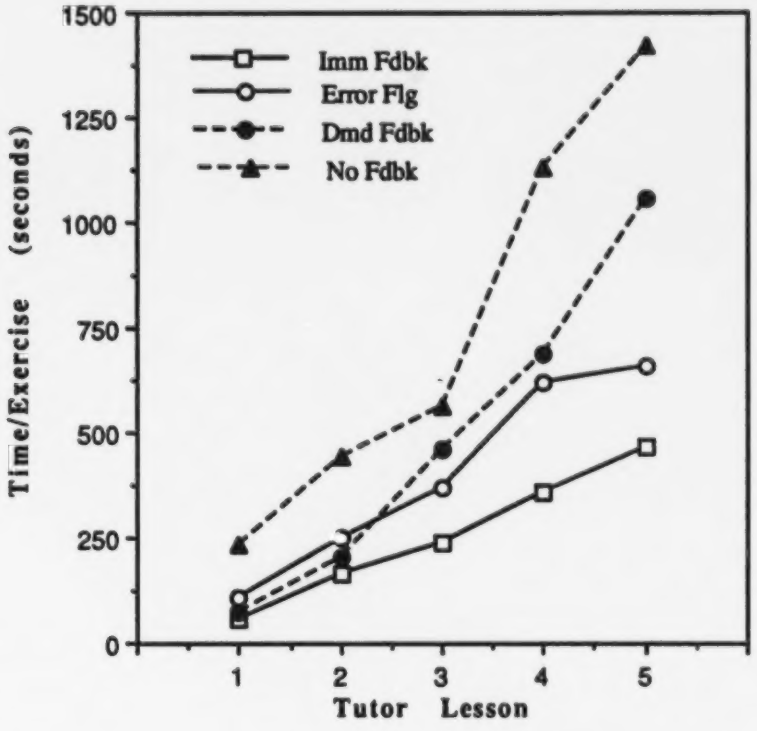

Figure 1. Mean Exercise Completion Time for the Five Tutor Lessons

Paper and Pencil Tests.

Results of the paper and pencil programming tests are displayed in Table 2. Two measures are reported for each test: (1) percent of exercises completed correctly and (2) number of production level errors. Overall, students in the three groups who received any feedback in learning are performing about equally well, while the students in the No Feedback condition are performing at a lower level. The feedback manipulation had a marginally significant effect on number of production errors in an analysis of variance ($F(3,36)=2.30$, $p<0.10$), while the effect on percent correct is non-significant.

Table 2

Paper and PencilPost Test Scores
% Exercises Correct
Mean Number of Errors

	Immediate Feedback	Error Flagging	Demand Feedback	No Feedback
Test 1				
%Correct	67%	62%	67%	51%
Errors	5.9	6.6	4.3	10.8
Test 2				
%Correct	45%	49%	50%	34%
Errors	33.2	22.1	19.5	52.6

OnLine Editor/Lisp Interpreter Test.

Results of the editor-based online test are displayed in Table 2. Two accuracy measures, percent of exercises completed correctly and number of production level errors are again reported. While this environment is most like the No Feedback practice environment, the No Feedback students are again performing worse than the the three groups that received feedback during learning. The feedback manipulation approaches significance for the number of production level errors , while it is nonsignificant for the percent correct measure. (A number of students exceeded the time limit for this test and failed to even begin the final exercise. In an analysis of just the first five exercises, there is a marginal effect of the feedback manipulation on number of production level errors). Elapsed time was also measured for this test. Interestingly, the two groups that had both coding freedom and feedback in practice completed this test somewhat more quickly than either the Immediate Feedback group which had no practice in self-correction or the No Tutor group.

Table 3
Editor/Lisp Window

% Exercises Correct
Mean Number of Errors
Elapsed Time (sec)

	Immediate Feedback	Error Flagging	Demand Feedback	No Feedback
%Correct	80%	73%	75%	63%
Errors	15.8	21.7	22.1	31.5
TotalTime	493	416	395	486

Immediate Feedback Tutor Test

Results of the Immediate Feedback Tutor posttest are displayed in Table 3. Students necessarily generated correct code in completing each of these exercises so the usual accuracy measures do not apply. However, we can look at performance as students complete the exercises. Table 3 displays three performance measures: (1) percent of goals completed correctly, (2) production firing time and (3) elapsed time. The first measure reports the probability that a student's first attempt is correct at each step in a solution. The second measure reports the average time to complete such a correct step. The third measure reports the mean time required to complete each exercise. As can be seen, differences between the groups are small, although again, students in the three feedback groups are performing slightly better than students in the No Feedback condition.

Table 3

Immediate Feedback PostTest

% Goals Correct
Production Firing Time (sec)
Elapsed Time (sec)

	Immediate Feedback	Error Flagging	Demand Feedback	No Feedback
%Correct	93%	91%	94%	90%
Prod Time	8.9	8.6	9.0	10.0
TotalTime	179	182	191	250

On balance, across three test environments, there is little performance difference across the three conditions in which feedback was presented. Students in the No Feedback condition, however, performed consistently worse than these groups. This suggests that as long as assistance is available while the student is working on the exercises and the student is required to generate correct solutions, the timing and control of the feedback does not have a strong effect on the degree of learning achieved. Students in the No Feedback condition were not required to achieve correct answers to the practice exercises (in fact, could not be required to in the absence of assistance). The tutor always verified the students' correct answers and displayed a correct answer when the answer was wrong, but this information was not sufficient to bring the No Feedback condition up to the level of the other three groups.

Questionnaires.

The results of the questionnaires presented after the second and fifth lessons are displayed in Table 4. The same seven questions were asked on both questionnaires. An eighth question, "How much would you like to learn more Lisp?" was included on the second questionnaire.

The most striking overall pattern is that there are virtually no reliable differences among the groups. The two questions for which there is a significant difference among the groups are displayed in bold.

The first two questions were intended to assess students' monitoring of their learning. The two low-feedback groups (Demand Feedback and No Feedback) generally found the exercises more difficult. The No Tutor group also showed signs of recognizing that they had learned the material less well. However, neither of these patterns was reliable.

Five of the questions asked students what they thought of the tutor. Again, there were virtually no differences among the groups. The similarity in responses is perhaps most surprising for the third question. The four tutor versions covered quite a wide range of tutor intrusiveness and a wide range of exercise difficulty, but overall students do not show a preference for one version over another. In an earlier study (Corbett & Anderson, 1990) students in the same four groups were asked this question after they had all switched to doing exercises with the standard immediate feedback tutor. Even with a common frame of reference, there were no differences in ratings of the four tutor versions.

There were two significant results in the second questionnaire. First, with the more difficult exercises in the last three lessons, students recognized that the more intrusive versions of the tutor were helping them complete the exercises more quickly. Second, there was a reliable difference among the groups when asked if they would like to learn more Lisp. This question was intended to get at whether students liked what they were doing, in the most general terms. This is of potential importance, since in a naturalistic environment the response may predict whether students stick with a course of learning. Note that the Immediate Feedback group responds most positively to this question, but the overall pattern is difficult to interpret, since the next highest rating comes from the group who received no feedback at all.

Table 4

Questionnaires

	Questionnaire 1				Questionnaire 2			
	Imm Fdbk	Err Flag	Dmd Fdbk	No Fdbk	Imm Fdbk	Err Flag	Dmd Fdbk	No Fdbk
1. How Difficult were the exercises? (1=Easy, 7=challenging)	3.3	2.8	3.8	4.1	5.2	4.9	5.7	5.7
2. How well did you learn the material? (1=Not Well, 7=Well)	5.8	5.3	5.7	5.3	4.8	5.3	5.1	4.5
3. How much did you like the tutor? (1=Disliked, 7=Liked)	5.1	4.8	5.7	4.9	5.5	4.6	4.3	4.5
4. Did the tutor help you finish more quickly? (1=Slower, 7=Faster)	5.3	4.2	4.7	4.2	**6.0**	**5.7**	**4.3**	**4.2**
5. Did the tutor help you understand better? (1=Interferred,7=Helped)	5.5	4.6	5.3	5.0	4.5	4.2	4.8	4.5
6. Did you like the tutor's assistance? (1=Disliked, 7=Liked)	5.3	4.4	5.5	4.8	5.2	4.7	4.3	4.5
7. Would you like more or less assistance? (1=Less, 7=More)	3.8	4.0	4.5	4.8	4.1	4.5	4.4	5.1
8.Would you like to learn more Lisp? (1=No, 7=Yes)					**6.5**	**5.0**	**4.9**	**5.9**

Summary and Conclusion

This study examined the effects on learning of four different feedback conditions. In three conditions help was available as the student worked while in the No Feedback condition no assistance was available. In two of the three feedback conditions, error feedback was presented immediately, while in the Feedback on Demand condition, feedback was presented only upon request of the student. Among the two conditions receiving immediate feedback, one group was given a feedback message and required to fix the error immediately. In the second, Error Flagging condition, errors were immediately marked in bold, but students were allowed to continue working as they pleased.

Students in the three groups who received assistance in learning performed about equally well and generally better than students in the No Feedback condition across posttests in three different environments. This suggests that as long as assistance is available in problem solving, and students are required to generate correct code, the specific feedback conditions may have little impact on rate of learning to program as measured relative the number of exercises completed. The manipulation did have an impact, of course, on rate of learning as measured in elapsed time. Students in the two immediate feedback conditions finished the exercises most quickly, followed by those in the Feedback on Demand condition. Students in the No Feedback condition took the longest by far. There is little or no evidence across posttest environments that the extra time expended in the Error Flagging or Feedback on Demand conditions compared, to the Immediate Feedback and Correction condition, led to the development of any useful skills. While these less intrusive conditions would be necessary for students to develop additional skills, e.g., debugging, they do not appear to be sufficient. It suggests that, in addition to giving students the freedom to develop such skill, explicit instruction in those skills is also desirable.

References

Anderson, J.R. (1983). *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Anderson, J.R. and Reiser, B.J. (1985) The LISP Tutor. *Byte, 10,* 159-175.

Corbett, A.T. and Anderson, J.R. (1989). Feedback timing and student control in the Lisp Intelligent Tutoring System. *The Proceedings of the Fourth International Conference on AI and Education,* Amsterdam.

Corbett, A.T. and Anderson, J.R. (1990). The effect of feedback control and learning to program with the Lisp Tutor. *The Proceedings of the Twelfth Annual Conference of the Cognitive Science Society,* Cambridge, MA.

Corbett, A.T. Anderson, J.R. and Patterson, E.G. (1990). Student modeling and tutoring flexibility in the Lisp Intelligent Tutoring System. In C. Frasson and G.Gauthier, (eds.) *Intelligent Tutoring Systems.* Norwoood, NJ: Ablex.

Kulik, J.A. and Kulik, C.C. (1988). Timing of feedback and verbal learning. *Review of Educational Research, 58,* 79-97.

Schmidt, R.A., Young, D.E., Swinnen, S. and Shapiro, D.C. (1989). Summary knowledge results for skill acquisition: Support for the guidance hypothesis. *Journal of Experimental Psychology: Learning, Memory and Cognition,* 15, 352-359.

Schooler, L.J. and Andsron, J.R. (1990) The disruptive potential of immediate feedback. *The Proceedings of the Twelfth Annual Conference of the Cognitive Science Society,* Cambridge, MA.